REMARKS

Claims 1-7, 9, 11, 13-23, and 28-31 are presented for consideration. Claims 1-4, 7, 9, 13, 15, 16, and 20 are amended. Claims 8, 10, 12, 24-27 are cancelled. Claims 28-31 are new.

Claims 1-7, 9-13, 15, and 17-27 were previously rejected under 35 U.S.C. § 103(a) as being unpatentable over Spyker et al. (U.S. Pat. 6,571,389), hereinafter referred to as Spyker, in view of Grate et al. (U.S. Pat. 5,956,483), hereinafter referred to as Grate, and further in view of Giroir et al. (U.S. Pat. 6,854,006), hereinafter referred to as Giroir. Claims 8, 14, and 16 were previously rejected under U.S.C. § 103(a) as being unpatentable over Spyker in view of Grate in view of Giroir and further in view of Schmidt et al. (U.S. Pat. 6,535,894), hereinafter referred to as Schmidt.

Claim 1 is amended to more clearly recite novel features of the present invention. Other existing features of claim 1 have been moved to new claims 28-31 to remove an unintended ambiguity.

Firstly, claim 1 is amended to more clearly recite that the application (i.e. program) being downloaded from a network, installed on a client device, and automatically launched on the client device is a stand-alone application that is independent of the network client application, i.e. web browser, and launched outside of the network client application. Claim 1 is further amended to clarify that the archive file that contains the selected stand-alone application, further contains a control module that plugs into the network client application, and which is granted greater operational permission within the client device than the network client application. Because of this greater permission, the control module has the ability to launch the stand-alone application outside of the network client application (i.e. outside of the web browser).

As it is known in the art, a network client application (i.e. a web browser) can activate network plug-in modules and applets, and may be able to open some other types of files, such as image files and word processing files, but network client applications are not permitted to launch independent (i.e. stand-alone) applications. Thus, there has previously not been available a method by which a stand alone application may be launched from within a network client application.

Spyker attempts to address this issue, but does so in a very different and limited way. In his prior art section, Spyker explains the benefits and drawbacks of Java applets versus Java applications. Spyker explains that Java applets are very easy to launch since they are launched within a web browser environment, which provides the appropriate operation environment for the Java applet. The Java applet is, however, limited to run only while the web browser environment is available (i.e. until the web browser is closed). Java applications, on the other hand, are executed outside of a web browser, but are very cumbersome to launch, requiring a user to manually setup their operating environment, which requires knowledge of JDK or JRE tools. Furthermore, Java applications cannot be launched automatically from within a web browser.

However, the Java engine needed to run a Java application is typically included with the application, while the Java engine for running an applet is set by the web browser. Thus, it is far easier to manage the appropriate Java engine when using an application than when using an applet.

Spyker thus, seeks to combine the ease of launching a Java applet with the ease of managing Java engines with a Java application. Spyker accomplishes this by first determining all the environmental settings provided by a web browser, and needed by an applet. The environmental settings are encapsulated into an installation program. It must be emphasized that like in the prior art, Spyker cannot launch his installation program from within a web browser. Indeed, Spyker's objective is to eliminate the need for a web browser. Thus, Spyker' installation program is itself a stand-alone program that is launched clicking on an icon (col. 9, lines 57-59). Once his stand-alone installation program is launched, it downloads and installs all the environmental settings needed by an applet, and which in his prior art would have been provided by a web browser. However in his case, since Spyker's installation program selects the environmental settings, it can control the choice of Java engine. After preparing the web-browser-like environment, Spyker launches his applet. Since the prepared environment mimics that of a web browser, the applet can be launched easily without having to prepare a JKE or JRE environment, as would have been needed by a Java Application. In this manner, Spyker combines the ease of launching a Java applet with the greater environmental control of a Java Application.

The point is, however, that Spyker teaches away from launching a program from within a web browser. Indeed, none of the cited prior art provides any method for launching a stand-alone application from within a web browser.

Giroir describes methods for using an applet to setup a telnet window to permit SNA network applications to communicate with IP network applications. As it would be understood, Giroir's applet is launched within the web browser, and remains under direct influence of the web browser.

Grate explains a manner by which a web browser may communicate with a stand-alone application running outside of the web browser. But Grate does not teach or suggest that the web browser launches the stand-alone application outside of the web browser. Indeed, Grate is silent on how his stand-alone application is launched. Thus, one must assume that Grate's stand-alone application is launched in the typical manner of clicking on a icon or using a command line execution.

New claims 28 and 29 further clarify the use of the term "parameter", as previously stated in claim 1. Basically, the term parameter is amended to recite "a plurality of user-available function tools" in order to better distinguish from environment parameters. The user-available function tools defined in claim 29 include tools for creating greeting cards, creating business cards, and creating a slide show presentation. Thus, the present stand-alone application will configure itself to provide a user with the selected user-accessible tools and to exclude from user access (or user view) unselected user-accessible tools. This provides for a user interface that is much easier to use do to the removal of functions that the user does not intend to use. For example, if a user wishes to create business cards, the presently claimed stand-alone application configures itself to provide the user only tools associated with the creation of business cards. Thus, the user need not be distracted by tool options associated with creating presentation files, or greeting cards, etc. None of the cited prior art teach or suggest an application capable of configuring itself to provide a user with only a subset of its functionality based on previously submitted user selection of his/her intended intentions.

The remaining pending claims are similarly amended to clarify and emphasize novel features as explained above.

In view of the foregoing amendments and remarks, Applicants respectfully request favorable reconsideration of the present application.

Respectfully submitted,

/Rosalio Haro/
Rosalio Haro
Registration No. 42,633

Please address all correspondence to:

Epson Research and Development, Inc.
Intellectual Property Department
150 River Oaks Parkway, Suite 225
San Jose, CA 95134
Phone: (408) 952-6131
Facsimile: (408) 954-9058
Customer No. 20178

Date: June 5, 2006